

#3

Docket No.: 57454-206

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Yasuhiro NUNOMURA

Serial No.:

Group Art Unit:

Filed: August 14, 2001

Examiner:

For: MULTIPROCESSOR SYSTEM CONTROLLING FREQUENCY OF CLOCK INPUT
TO PROCESSOR ACCORDING TO RATIO OF PROCESSING TIMES OF
PROCESSORS, AND METHOD THEREOF

**CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Commissioner for Patents
Washington, DC 20231

Sir:

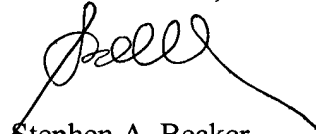
In accordance with the provisions of 35 U.S.C. 119, Applicant hereby claims the priority of:

Japanese Patent Application No. 2001-009802, filed January 18, 2001

cited in the Declaration of the present application. A Certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY



Stephen A. Becker
Registration No. 26,527

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 SAB:prp
Date: August 14, 2001
Facsimile: (202) 756-8087

57454-206
NUNOMURA
August 14, 2001

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2001年 1月18日

出 願 番 号

Application Number:

特願2001-009802

出 願 人

Applicant(s):

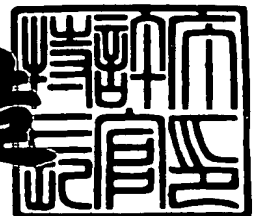
三菱電機株式会社



2001年 2月 9日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3006741

【書類名】 特許願

【整理番号】 528095JP01

【提出日】 平成13年 1月18日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/16
G06F 1/32
G06F 1/04

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

【氏名】 布村 泰浩

【特許出願人】

【識別番号】 000006013

【氏名又は名称】 三菱電機株式会社

【代理人】

【識別番号】 100064746

【弁理士】

【氏名又は名称】 深見 久郎

【選任した代理人】

【識別番号】 100085132

【弁理士】

【氏名又は名称】 森田 俊雄

【選任した代理人】

【識別番号】 100091409

【弁理士】

【氏名又は名称】 伊藤 英彦

【選任した代理人】

【識別番号】 100096781

【弁理士】

【氏名又は名称】 堀井 豊

【選任した代理人】

【識別番号】 100096792

【弁理士】

【氏名又は名称】 森下 八郎

【手数料の表示】

【予納台帳番号】 008693

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 マルチプロセッサシステムおよびその制御方法

【特許請求の範囲】

【請求項 1】 複数のプロセッサが並列に処理を行なった後に、同期を取るマルチプロセッサシステムであって、

第 1 のプロセッサと、

前記第 1 のプロセッサと異なる第 2 のプロセッサと、

前記第 1 のプロセッサの処理時間と前記第 2 のプロセッサの処理時間との比率に応じて、前記第 1 のプロセッサに入力されるクロックの周波数を制御するためのクロック周波数制御手段とを含む、マルチプロセッサシステム。

【請求項 2】 前記マルチプロセッサシステムはさらに、前記第 1 のプロセッサに入力されるクロックの周波数に応じて、前記第 1 のプロセッサに供給される電源電圧を制御するための電圧制御手段を含む、請求項 1 記載のマルチプロセッサシステム。

【請求項 3】 前記マルチプロセッサシステムはさらに、前記第 1 のプロセッサの処理時間見積値と前記第 2 のプロセッサの処理時間見積値とが予め格納される格納手段と、

前記格納手段に格納された第 1 のプロセッサの処理時間見積値と第 2 のプロセッサの処理時間見積値との比率に応じて前記クロック周波数制御手段を制御して、前記第 1 のプロセッサに入力されるクロックの周波数を変更するための周波数決定手段とを含む、請求項 1 または 2 記載のマルチプロセッサシステム。

【請求項 4】 前記マルチプロセッサシステムはさらに、前記第 1 のプロセッサの処理時間見積値と前記第 2 のプロセッサの処理時間見積値との比率が予め格納される格納手段を含み、

前記格納手段に格納された比率に応じて前記クロック周波数制御手段を制御して、前記第 1 のプロセッサに入力されるクロックの周波数を変更するための周波数決定手段とを含む、請求項 1 または 2 記載のマルチプロセッサシステム。

【請求項 5】 前記マルチプロセッサシステムはさらに、前回の処理において、前記第 1 のプロセッサの処理が前記第 2 のプロセッサの処理よりも先に完了

した場合には、前記クロック周波数制御手段に第 1 のプロセッサに入力されるクロックの周波数を低くさせ、

前回の処理において、前記第 2 のプロセッサの処理が前記第 1 のプロセッサの処理よりも先に完了した場合には、前記第 1 のプロセッサに入力されるクロックの周波数を高くさせるための周波数決定手段を含む、請求項 1 または 2 記載のマルチプロセッサシステム。

【請求項 6】 前記マルチプロセッサシステムはさらに、前記第 1 のプロセッサの処理と前記第 2 のプロセッサの処理とのいずれが先に完了するかを予測するための予測手段と、

前記予測手段による予測結果に応じて前記クロック周波数制御手段を制御して、前記第 1 のプロセッサに入力されるクロックの周波数を変更するための周波数決定手段とを含む、請求項 1 または 2 記載のマルチプロセッサシステム。

【請求項 7】 前記予測手段は、前回の処理において、前記第 1 のプロセッサの処理が前記第 2 のプロセッサの処理よりも先に完了した場合には、前記第 1 のプロセッサの方が先に処理を完了すると予測し、

前回の処理において、前記第 2 のプロセッサの処理が前記第 1 のプロセッサの処理よりも先に完了した場合には、前記第 2 のプロセッサの方が先に処理を完了すると予測する、請求項 6 記載のマルチプロセッサシステム。

【請求項 8】 前記予測手段は、前回の処理において、前記第 1 のプロセッサの処理が前記第 2 のプロセッサの処理よりも先に完了した場合には、前記第 2 のプロセッサの方が先に処理を完了すると予測し、

前回の処理において、前記第 2 のプロセッサの処理が前記第 1 のプロセッサの処理よりも先に完了した場合には、前記第 1 のプロセッサの方が先に処理を完了すると予測する、請求項 6 記載のマルチプロセッサシステム。

【請求項 9】 前記マルチプロセッサシステムはさらに、前記第 1 のプロセッサの処理時間を計測するための第 1 の計測手段と、

前記第 2 のプロセッサの処理時間を計測するための第 2 の計測手段と、

前記第 1 の計測手段による計測結果および前記第 2 の計測手段による計測結果に応じて前記クロック周波数制御手段を制御して、前記第 1 のプロセッサに入力

されるクロックの周波数を変更するための周波数決定手段とを含む、請求項 1 または 2 記載のマルチプロセッサシステム。

【請求項 1 0】 複数のプロセッサが並列に処理を行なった後に、同期を取るマルチプロセッサシステムの制御方法であって、

第 1 のプロセッサの処理時間と第 2 のプロセッサの処理時間との比率を算出するステップと、

前記算出された比率に応じて、前記第 1 のプロセッサに入力されるクロックの周波数を制御するステップとを含む、マルチプロセッサシステムの制御方法。

【請求項 1 1】 前記制御方法はさらに、前記第 1 のプロセッサに入力されるクロックの周波数に応じて、前記第 1 のプロセッサに供給される電源電圧を制御するステップを含む、請求項 1 0 記載のマルチプロセッサシステムの制御方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、複数のプロセッサを接続して高速に処理を行なう並列処理技術に関し、特に、消費電力を削減することが可能なマルチプロセッサシステムおよびその制御方法に関する。

【0 0 0 2】

【従来の技術】

近年、マルチメディア処理や高精細画像処理などのさまざまな分野で、プロセッサ性能の向上に対する要求が高まっている。しかし、現在の L S I (Large Scale Integrated circuit) 製造技術ではデバイスの高速化にも限度がある。そこで、並列処理が注目を浴びており、マルチプロセッサシステムに関する研究、開発が盛んに行なわれている。

【0 0 0 3】

一般に、複数のプロセッサが並列処理で 1 つの処理を行なう場合、できるだけ個々のプロセッサに均等な負荷分散を行なって、システム全体の性能を最大限に引き出す必要がある。しかし、実際のアプリケーション設計において、開発を容

易にしたり、システムコストの削減を考慮したりすると、必ずしも均等な負荷分散ができる訳ではない。したがって、一方のプロセッサが他方のプロセッサの処理完了を待たなければならない状況が多々発生する。

【 0 0 0 4 】

一方、携帯機器に搭載された電池の寿命の向上や、近年盛んに唱えられている環境への配慮などのために、システムの低消費電力化が要求されている。このようなシステムの低消費電力化の一手法として、システムの稼動状況に応じてシステム全体や一部のクロック周波数を低減して、消費電力を抑えることを挙げることができる。

【 0 0 0 5 】

図 1 7 は、従来のマルチプロセッサシステムにおける低消費電力化の一例を示す図である。このマルチプロセッサシステムは、プロセッサ A 1 0 1 と、プロセッサ B 1 0 2 と、プログラム A が格納されたメモリ 1 0 3 と、プログラム B が格納されたメモリ 1 0 4 と、プロセッサ A 1 0 1 に入力されるクロックの供給／切断を制御するクロック供給制御部 1 0 5 とを含む。

【 0 0 0 6 】

プロセッサ A 1 0 1 およびプロセッサ B 1 0 2 がそれぞれ、メモリ 1 0 3 に格納されたプログラム A およびメモリ 1 0 4 に格納されたプログラム B を実行する場合において、プロセッサ A 1 0 1 の処理時間の方がプロセッサ B 1 0 2 の処理時間よりも短いとする。このとき、プロセッサ A 1 0 1 がプログラム A の最後で、プロセッサ B 1 0 2 の処理完了を待つことによって、2 つのプロセッサの同期が取られていた。

【 0 0 0 7 】

このようなシステムにおいて、プロセッサ B 1 0 2 が処理を完了するまで、プロセッサ A 1 0 1 が何も実行する必要がない場合、クロック供給制御部 1 0 5 にクロック制御指令を発行し、プロセッサ A 1 0 1 に供給されるクロックを停止して、プロセッサ A 1 0 1 の消費電力を削減していた。

【 0 0 0 8 】

【発明が解決しようとする課題】

しかし、上述したマルチプロセッサシステムにおいて、プロセッサ A 1 0 1 が処理を完了してから、プロセッサ B 1 0 2 が処理を完了するまでの間、プロセッサ A 1 0 1 のクロックを停止して消費電力を削減することは可能であるが、さらなる消費電力の削減を行なうことができないという問題点があった。

【0 0 0 9】

本発明は、上記問題点を解決するためになされたものであり、第 1 の目的は、消費電力を大きく削減することが可能なマルチプロセッサシステムおよびその制御方法を提供することである。

【0 0 1 0】

第 2 の目的は、消費電力を削減する際の処理を効率的に行なうことが可能なマルチプロセッサシステムおよびその制御方法を提供することである。

【0 0 1 1】

【課題を解決するための手段】

請求項 1 に記載のマルチプロセッサシステムは、複数のプロセッサが並列に処理を行なった後に、同期を取るマルチプロセッサシステムであって、第 1 のプロセッサと、第 1 のプロセッサと異なる第 2 のプロセッサと、第 1 のプロセッサの処理時間と第 2 のプロセッサの処理時間との比率に応じて、第 1 のプロセッサに入力されるクロックの周波数を制御するためのクロック周波数制御手段とを含む。

【0 0 1 2】

クロック周波数制御手段は、第 1 のプロセッサの処理時間と第 2 のプロセッサの処理時間との比率に応じて、第 1 のプロセッサに入力されるクロックの周波数を制御するので、第 1 のプロセッサに入力されるクロックの周波数を最適にすることができ、第 1 のプロセッサの消費電力を削減することが可能となる。

【0 0 1 3】

請求項 2 に記載のマルチプロセッサシステムは、請求項 1 に記載のマルチプロセッサシステムであって、さらに第 1 のプロセッサに入力されるクロックの周波数に応じて、第 1 のプロセッサに供給される電源電圧を制御するための電圧制御手段を含む。

【 0 0 1 4 】

第 1 のプロセッサに入力されるクロックの周波数の低減に応じて、電源電圧を下げるができるため、さらに第 1 のプロセッサの消費電力を削減することが可能となる。

【 0 0 1 5 】

請求項 3 に記載のマルチプロセッサシステムは、請求項 1 または 2 記載のマルチプロセッサシステムであって、さらに第 1 のプロセッサの処理時間見積値と第 2 のプロセッサの処理時間見積値とが予め格納される格納手段と、格納手段に格納された第 1 のプロセッサの処理時間見積値と第 2 のプロセッサの処理時間見積値との比率に応じてクロック周波数制御手段を制御して、第 1 のプロセッサに入力されるクロックの周波数を変更するための周波数決定手段とを含む。

【 0 0 1 6 】

周波数決定手段は、第 1 のプロセッサの処理時間見積値と第 2 のプロセッサの処理時間見積値との比率に応じて、第 1 のプロセッサに入力されるクロックの周波数を変更するので、クロック周波数を変更するための処理を簡略化することが可能となる。

【 0 0 1 7 】

請求項 4 に記載のマルチプロセッサシステムは、請求項 1 または 2 記載のマルチプロセッサシステムであって、さらに第 1 のプロセッサの処理時間見積値と第 2 のプロセッサの処理時間見積値との比率が予め格納される格納手段を含み、格納手段に格納された比率に応じてクロック周波数制御手段を制御して、第 1 のプロセッサに入力されるクロックの周波数を変更するための周波数決定手段とを含む。

【 0 0 1 8 】

周波数決定手段は、格納手段に格納された比率に応じて、第 1 のプロセッサに入力されるクロックの周波数を変更するので、クロック周波数を変更するための処理を簡略化することが可能となる。

【 0 0 1 9 】

請求項 5 に記載のマルチプロセッサシステムは、請求項 1 または 2 記載のマル

チプロセッサシステムであって、さらに前回の処理において、第1のプロセッサの処理が第2のプロセッサの処理よりも先に完了した場合には、クロック周波数制御手段に第1のプロセッサに入力されるクロックの周波数を低くさせ、前回の処理において、第2のプロセッサの処理が第1のプロセッサの処理よりも先に完了した場合には、第1のプロセッサに入力されるクロックの周波数を高くさせるための周波数決定手段を含む。

【 0 0 2 0 】

周波数決定手段は、前回の第1のプロセッサおよび第2のプロセッサの処理状況に応じて、第1のプロセッサに入力されるクロックの周波数を変更するので、プロセッサの処理時間が実行毎に異なるような場合であっても、最適なクロック周波数を決定することが可能となる。

【 0 0 2 1 】

請求項6に記載のマルチプロセッサシステムは、請求項1または2記載のマルチプロセッサシステムであって、さらに第1のプロセッサの処理と第2のプロセッサの処理とのいずれが先に完了するかを予測するための予測手段と、予測手段による予測結果に応じてクロック周波数制御手段を制御して、第1のプロセッサに入力されるクロックの周波数を変更するための周波数決定手段とを含む。

【 0 0 2 2 】

周波数決定手段は、予測手段による予測結果に応じて、第1のプロセッサに入力されるクロックの周波数を変更するので、最適なクロック周波数を決定することが可能となる。

【 0 0 2 3 】

請求項7に記載のマルチプロセッサシステムは、請求項6記載のマルチプロセッサシステムであって、予測手段は、前回の処理において、第1のプロセッサの処理が第2のプロセッサの処理よりも先に完了した場合には、第1のプロセッサの方が先に処理を完了すると予測し、前回の処理において、第2のプロセッサの処理が第1のプロセッサの処理よりも先に完了した場合には、第2のプロセッサの方が先に処理を完了すると予測する。

【 0 0 2 4 】

したがって、第1のプロセッサが早く処理を完了したら、クロック周波数を低くしても、やはり第1のプロセッサが早く処理を完了する傾向にあるという戦略を表わすことができ、最適なクロック周波数を決定することが可能となる。

【0025】

請求項8に記載のマルチプロセッサシステムは、請求項6記載のマルチプロセッサシステムであって、予測手段は、前回の処理において、第1のプロセッサの処理が第2のプロセッサの処理よりも先に完了した場合には、第2のプロセッサの方が先に処理を完了すると予測し、前回の処理において、第2のプロセッサの処理が第1のプロセッサの処理よりも先に完了した場合には、第1のプロセッサの方が先に処理を完了すると予測する。

【0026】

したがって、第1のプロセッサが早く処理を完了したら、クロック周波数を低くすることになるので、次は第2のプロセッサが早く処理を完了する傾向にあるという戦略を表わすことができ、最適なクロック周波数を決定することが可能となる。

【0027】

請求項9に記載のマルチプロセッサシステムは、請求項1または2記載のマルチプロセッサシステムであって、さらに第1のプロセッサの処理時間を計測するための第1の計測手段と、第2のプロセッサの処理時間を計測するための第2の計測手段と、第1の計測手段による計測結果および第2の計測手段による計測結果に応じてクロック周波数制御手段を制御して、第1のプロセッサに入力されるクロックの周波数を変更するための周波数決定手段とを含む。

【0028】

周波数決定手段は、第1の計測手段による計測結果および第2の計測手段による計測結果に応じてクロック周波数制御手段を制御するので、最適なクロック周波数を決定することが可能となる。

【0029】

請求項10に記載のマルチプロセッサシステムの制御方法は、複数のプロセッサが並列に処理を行なった後に、同期を取るマルチプロセッサシステムの制御方

法であって、第 1 のプロセッサの処理時間と第 2 のプロセッサの処理時間との比率を算出するステップと、算出された比率に応じて、第 1 のプロセッサに入力されるクロックの周波数を制御するステップとを含む。

【0030】

第 1 のプロセッサの処理時間と第 2 のプロセッサの処理時間との比率に応じて、第 1 のプロセッサに入力されるクロックの周波数を制御するので、第 1 のプロセッサに入力されるクロックの周波数を最適にすることができ、第 1 のプロセッサの消費電力を削減することが可能となる。

【0031】

請求項 11 に記載のマルチプロセッサシステムの制御方法は、請求項 10 記載のマルチプロセッサの制御方法であって、さらに第 1 のプロセッサに入力されるクロックの周波数に応じて、第 1 のプロセッサに供給される電源電圧を制御するステップを含む。

【0032】

第 1 のプロセッサに入力されるクロックの周波数の低減に応じて、電源電圧を下げるため、さらに第 1 のプロセッサの消費電力を削減することが可能となる。

【0033】

【発明の実施の形態】

（実施の形態 1）

図 1 は、本発明の実施の形態 1 におけるマルチプロセッサシステムの概略構成を示すブロック図である。このマルチプロセッサシステムは、プロセッサ A 11 と、プロセッサ B 12 と、プログラム A およびプログラム A の処理時間見積値 T_a が格納されたメモリ 13 と、プログラム B およびプログラム B の処理時間見積値 T_b が格納されたメモリ 14 と、プロセッサ A 11 に入力されるクロック周波数を制御するクロック周波数制御部 15 と、プロセッサ A 11 に電源電圧を供給する可変電源 16 とを含む。

【0034】

プロセッサ A 11 が、メモリ 13 に格納されたプログラム A を実行し、プロセ

ッサ B 1 2 が、メモリ 1 4 に格納されたプログラム B を実行することによって、並列に処理が行なわれる。このシステムは、プロセッサ A 1 1 とプロセッサ B 1 2 との間で情報を通信するための通信機能を有しており、たとえば、割込み、メモリマップド I/O (Input/Output)、共有メモリ等によって実現される。この通信機能は、プロセッサ A 1 1 がプロセッサ B 1 2 に対してプログラム B の実行を起動する際、およびプロセッサ B 1 2 がプログラム B の処理を完了したことをプロセッサ A 1 1 に対して通知する際に使用される。

【 0 0 3 5 】

プロセッサ A 1 1 は、メモリ 1 3 に格納されたプログラム A の処理時間見積値 T a およびメモリ 1 4 に格納されたプログラム B の処理時間見積値 T b を参照して、プロセッサ A 1 1 に入力されるクロック周波数を決定し、クロック周波数制御部 1 5 にクロック制御指令を発行する。プログラム A の処理時間見積値 T a およびプログラム B の処理時間見積値 T b は、たとえば、プログラム A、B を作成するプログラマによって見積もられた値であり、予めメモリ 1 3 およびメモリ 1 4 に格納されている。なお、クロック周波数の決定方法の詳細は後述する。

【 0 0 3 6 】

図 2 は、クロック周波数制御部 1 5 の詳細を説明するための図である。このクロック周波数制御部 1 5 は、発振回路 1 7 と、発振回路 1 7 から出力されるクロックを分周する複数の分周器 1 8 - 1 ~ 1 8 - m と、発振回路 1 7 および複数の分周器 1 8 - 1 ~ 1 8 - m の出力のいずれかを選択して出力するセレクタ 1 9 とを含む。

【 0 0 3 7 】

発振回路 1 7 は、プロセッサ A 1 1 に入力されるクロックの最高周波数以上のクロックを出力する。分周器 1 8 - 1 ~ 1 8 - m は、それぞれ発振回路 1 7 から出力されるクロックを $1/2^i$ ($i = 1, 2, \dots, m$) に分周する。セレクタ 1 9 は、プロセッサ A 1 1 から出力されるクロック制御指令（周波数選択信号）に応じて、発振回路 1 7 および分周器 1 8 - 1 ~ 1 8 - m からの出力を選択して出力する。

【 0 0 3 8 】

図 3 は、本実施の形態におけるプロセッサ A 1 1 の処理手順を説明するためのフローチャートである。まず、プロセッサ A 1 1 は、プロセッサ B 1 2 に対して起動要求を出力して、プロセッサ B 1 2 にプログラム B の実行を要求する (S 1 1)。そして、プロセッサ A 1 1 は、メモリ 1 3 に格納されたプログラム A の処理時間見積値 T a およびメモリ 1 4 に格納されたプログラム B の処理時間見積値 T b を参照し (S 1 2)、見積値 T b が見積値 T a よりも大きいか否かを判定する (S 1 3)。見積値 T b が見積値 T a 以下であれば (S 1 3, N o)、プロセッサ A 1 1 のクロック周波数を変更せずに、ステップ S 1 6 へ処理が進む。

【 0 0 3 9 】

また、見積値 T b が見積値 T a よりも大きければ (S 1 3, Y e s)、プロセッサ A 1 1 は、プログラム A の処理時間見積値 T a およびプログラム B の処理時間見積値 T b から、次式に示す範囲内において、プロセッサ A 1 1 に新たに入力されるクロック周波数を決定して設定する (S 1 4)。なお、f は元のクロック周波数を、f' は新たなクロック周波数を示している。

【 0 0 4 0 】

$$f \times T a / T b \leq f' < f \quad \cdots (1)$$

たとえば、 $T b / 4 < T a \leq T b / 2$ の場合には、1 / 2 分周器 1 8 - 1 から出力されるクロックが選択されるように、プロセッサ A 1 1 がセレクタ 1 9 にクロック制御指令を出力する。また、 $T b / 8 < T a \leq T b / 4$ の場合には、1 / 4 分周器 1 8 - 2 から出力されるクロックが選択されるように、プロセッサ A 1 1 がセレクタ 1 9 にクロック制御指令を出力する。

【 0 0 4 1 】

次に、プロセッサ A 1 1 は、可変電源 1 6 を制御してプロセッサ A 1 1 に供給される電源電圧を下げ (S 1 5)、プログラム A の実行を開始する (S 1 6)。プロセッサ A 1 1 は、プログラム A の実行を完了すると、上述した通信機能を用いてプロセッサ B 1 2 から処理完了の通知があるか否かを判定する (S 1 7)。プロセッサ B 1 2 から処理完了の通知がなければ (S 1 7, N o)、処理完了の通知があるまで待機する。

【 0 0 4 2 】

また、プロセッサ B 1 2 から処理完了の通知があれば (S 1 7, Y e s)、プロセッサ A 1 1 は、可変電源 1 6 を制御してプロセッサ A 1 1 に供給される電源電圧を元に戻す。そして、プロセッサ A 1 1 はクロック周波数制御部 1 5 に対してクロック制御指令を発行して、クロック周波数を元のクロック周波数に戻して (S 1 8)、処理を終了する。

【 0 0 4 3 】

なお、このようにクロック周波数が低くなると電源電圧を低くすることができるのは、クロック周波数が低くなると各信号の遷移時間 (遅延時間) が多少小さくなくても回路動作に支障が生じないためである。クロック周波数と許容できる電圧値との関係は予め実験等によって定めることができる。上述した可変電源 1 6 の制御は、予めクロック周波数に対応した電源電圧値をテーブルに格納しておき、プロセッサ A 1 1 がそのテーブルを参照することによって行なわれる。

【 0 0 4 4 】

図 4 は、図 3 に示すプロセッサ A 1 1 の処理手順を、C 言語風の疑似プログラムによって表わしたものである。このプログラムにおいて、“B _ i s _ d o n e” はプロセッサ B 1 2 によるプログラム B の実行状態を示しており、“0” のときはプログラム B の実行中であることを示し、“1” のときはプログラム B の実行が完了していることを示している。なお、各ステップの処理内容は、コメント文で詳細に説明されているので、ここでの説明は行なわない。

【 0 0 4 5 】

図 4 に示すプログラムにおいては、プロセッサ B 1 2 からのプログラム B の完了通知は、割込みによって行なわれている。割込み以外によって通信機能を実現する場合には、上述したメモリマップド I / O または共有メモリをポーリングすることによって、プロセッサ B 1 2 からのプログラム B の完了通知を得ることができる。

【 0 0 4 6 】

以上説明したように、本実施の形態におけるマルチプロセッサシステムによれば、プロセッサ A 1 1 とプロセッサ B 1 2 とが並列に処理を実行する際、プロセッサ A 1 1 がプロセッサ B 1 2 よりも先に処理が完了する場合には、プロセッサ

A 1 1 を低速動作させるようにした。したがって、CMOS (Complementary Metal Oxide Semiconductor) 論理回路の消費電力がクロック周波数に比例することから、プロセッサ A 1 1 の消費電力を削減することが可能となった。

【0 0 4 7】

また、プロセッサ A 1 1 を低速動作させることにより、プロセッサ A 1 1 の電源電圧を下げるができるようになった。したがって、CMOS 論理回路の消費電力が電源電圧の 2 乗に比例することから、プロセッサ A 1 1 の消費電力をさらに削減することが可能となった。

【0 0 4 8】

(実施の形態 2)

図 5 は、本発明の実施の形態 2 におけるマルチプロセッサシステムの概略構成を示すブロック図である。本実施の形態におけるマルチプロセッサシステムは、図 1 に示す実施の形態 1 におけるマルチプロセッサシステムと比較して、メモリ 1 3 および 1 4 のそれぞれにプログラム A の処理時間見積値 T_a およびプログラム B の処理時間見積値 T_b が予め格納される代わりに、メモリ 1 3 にプログラム A と B との処理時間比率 $R (= T_a / T_b)$ が予め格納されている点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。なお、処理時間比率 R は、たとえば、プログラム A、B を作成するプログラマによって見積もられた値であり、予めメモリ 1 3 に格納されている。

【0 0 4 9】

図 6 は、本実施の形態におけるプロセッサ A 1 1 の処理手順を説明するためのフローチャートである。本実施の形態におけるプロセッサ A 1 1 の処理手順は、図 3 に示す実施の形態 1 におけるプロセッサ A 1 1 の処理手順と比較して、ステップ S 1 2 および S 1 3 がステップ S 2 2 に置換されている点のみが異なる。したがって、重複する処理手順の詳細な説明は繰返さない。

【0 0 5 0】

プロセッサ A 1 1 は、プロセッサ B 1 2 に起動要求を出力し (S 1 1)、メモリ 1 3 に格納されたプログラム A と B との処理時間比率 R を参照する (S 2 2)。そして、プロセッサ A 1 1 は、次式に示す範囲内において、プロセッサ A 1 1

に新たに入力されるクロック周波数を決定して設定する（S 1 4）。なお、 f は元のクロック周波数を、 f' は新たなクロック周波数を示している。

【0 0 5 1】

$$R \leq f' / f < 1 \quad \dots (2)$$

たとえば、 $1/4 < R \leq 1/2$ の場合には、 $1/2$ 分周器 1 8 - 1 から出力されるクロックが選択されるように、プロセッサ A 1 1 がセクタ 1 9 にクロック制御指令を出力する。また、 $1/8 < R \leq 1/4$ の場合には、 $1/4$ 分周器 1 8 - 2 から出力されるクロックが選択されるように、プロセッサ A 1 1 がセクタ 1 9 にクロック制御指令を出力する。

【0 0 5 2】

以上説明したように、本実施の形態におけるマルチプロセッサシステムによれば、実施の形態 1 において説明した効果に加えて、新たなクロック周波数を決定するための処理が簡略化されて、プロセッサ A 1 1 の処理負担を軽減することが可能となった。なお、本実施の形態におけるマルチプロセッサシステムは、プログラム A がプログラム B によって実現される機能と呼出す場合であって、常にプログラム A のみがプログラム B を呼出すようにプログラミングされている場合に有効である。

【0 0 5 3】

（実施の形態 3）

図 7 は、本発明の実施の形態 3 におけるマルチプロセッサシステムの概略構成を示すブロック図である。本実施の形態におけるマルチプロセッサシステムは、図 1 に示す実施の形態 1 におけるマルチプロセッサシステムと比較して、クロック周波数制御部 1 5 からのクロック信号および可変電源 1 6 からの電源電圧がプロセッサ B 1 2 に供給される点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。

【0 0 5 4】

図 8 は、本実施の形態におけるプロセッサ A 1 1 の処理手順を説明するためのフローチャートである。本実施の形態におけるプロセッサ A 1 1 の処理手順は、図 3 に示す実施の形態 1 におけるプロセッサ A 1 1 の処理手順と比較して、ステ

ップ S 1 4 および S 1 5 がステップ S 3 4 および S 3 5 に置換されている点のみが異なる。したがって、重複する処理手順の詳細な説明は繰返さない。

【 0 0 5 5 】

ステップ S 1 3 において、見積値 T b が見積値 T a よりも大きければ (S 1 3 , Y e s) 、プロセッサ A 1 1 は、可変電源 1 6 を制御してプロセッサ A 1 1 に供給される電源電圧を上げる (S 3 4) 。

【 0 0 5 6 】

次に、プロセッサ A 1 1 は、プログラム A の処理時間見積値 T a およびプログラム B の処理時間見積値 T b から、次式に示す範囲内において、プロセッサ B 1 2 に新たに入力されるクロック周波数を決定して設定する (S 3 5) 。なお、 f は元のクロック周波数を、 f ' は新たなクロック周波数を示している。また、元のクロック周波数 f は、 1 / 1 6 分周器 1 8 - 4 からの出力が選択されている場合のクロック周波数であるとする。

【 0 0 5 7 】

$$f < f' \leq f \times T b / T a \quad \cdots (3)$$

たとえば、 $2 T a \leq T b < 4 T a$ の場合には、 1 / 8 分周器 1 8 - 3 から出力されるクロック (元のクロック周波数 f の 2 倍) が選択されるように、プロセッサ A 1 1 がセクタ 1 9 にクロック制御指令を出力する。また、 $4 T a \leq T b < 8 T a$ の場合には、 1 / 4 分周器 1 8 - 2 から出力されるクロック (元のクロック周波数 f の 4 倍) が選択されるように、プロセッサ A 1 1 がセクタ 1 9 にクロック制御指令を出力する。そして、ステップ S 1 6 以降の処理を行なう。

【 0 0 5 8 】

以上説明したように、本実施の形態におけるマルチプロセッサシステムによれば、プロセッサ A 1 1 とプロセッサ B 1 2 とが並列に処理を実行する際、プロセッサ A 1 1 がプロセッサ B 1 2 よりも先に処理が完了する場合にのみ、プロセッサ B 1 2 を高速動作させるようにした。したがって、 CMOS 論理回路の消費電力がクロック周波数に比例することから、プロセッサ B 1 2 の通常の処理状態 (高速動作させない状態) における消費電力を削減することが可能となった。

【 0 0 5 9 】

また、プロセッサ B 1 2 の通常の処理状態において低速動作させることにより、プロセッサ B 1 2 の電源電圧を下げるできるようになった。したがって、CMOS 論理回路の消費電力が電源電圧の 2 乗に比例することから、プロセッサ B 1 2 の通常状態における消費電力をさらに削減することが可能となった。なお、本実施の形態におけるマルチプロセッサシステムは、予めマルチプロセッサシステム全体を低速動作させておき、必要に応じて処理の一部でプロセッサを高速に動作させるようなシステムに有効である。

【0060】

(実施の形態 4)

実施の形態 1 ～ 3 のマルチプロセッサシステムにおいては、プログラマが予め、プログラム A の処理時間見積値 T_a とプログラム B の処理時間見積値 T_b とを決めておき、その値に応じてクロック周波数や電源電圧を変更して低消費電力化を実現していた。しかし、プログラム A の処理時間やプログラム B の処理時間が実行毎に異なるような場合には、実施の形態 1 ～ 3 のマルチプロセッサシステムを構築することができない。

【0061】

本発明の実施の形態 4 におけるマルチプロセッサシステムは、プログラム A とプログラム B とが、プロセッサ A 1 1 とプロセッサ B 1 2 とによって繰返し並列に実行されるような場合に、前回の処理においてプロセッサ A 1 1 がプロセッサ B 1 2 からのプログラム B の処理完了通知を待ったか否かによって、クロック周波数を変更するものである。

【0062】

図 9 は、本発明の実施の形態 4 におけるマルチプロセッサシステムの概略構成を示すブロック図である。本実施の形態におけるマルチプロセッサシステムは、図 1 に示す実施の形態 1 におけるマルチプロセッサシステムと比較して、メモリ 1 3 および 1 4 のそれぞれにプログラム A の処理時間見積値 T_a およびプログラム B の処理時間見積値 T_b が格納されていない点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。なお、図 2 に示す実施の形態 1 におけるクロック制御部 1 5 は、発振回路 1 7 および複数の分周器 1 8 - 1 ～

18-mから出力されるクロックを切替えて、 $1/2^i$ 分周されたクロックを出力するものであったが、本実施の形態におけるクロック制御部はさらに細かく段階的にクロックを切替えることができるものとする。

【0063】

図10は、本発明の実施の形態4におけるプロセッサA11の処理手順を説明するためのフローチャートである。まず、プロセッサA11は、プロセッサB12に対して起動要求を出力して、プロセッサB12にプログラムBの実行を要求する(S41)。そして、プロセッサA11は、前回の処理に対応する情報I(n-1)を参照し(S42)、情報I(n-1)と予め定められたしきい値WAIT_LIMITとを比較して、前回のプログラムAの処理においてプロセッサB12からの完了通知を待ったか否かを判定する(S43)。

【0064】

プロセッサA11が、プロセッサB12からの完了通知を待った場合、すなわち情報I(n-1)がしきい値WAIT_LIMITよりも大きい場合には(S43, Yes)、クロック周波数制御部15を制御して、クロック周波数を1段だけ遅くする(S44)。また、プロセッサA11が、プロセッサB12からの完了通知を待たなかった場合、すなわち情報I(n-1)がしきい値WAIT_LIMIT以下の場合には(S43, No)、プロセッサA11がプログラムAの処理を完了する前に、プロセッサB12が処理を完了していたか否かを判定する(S45)。

【0065】

プロセッサA11がプログラムAの処理を完了する前に、プロセッサB12が処理を完了していた場合、すなわち情報I(n-1)が“0”の場合には(S45, Yes)、クロック制御部15を制御して、クロック周波数を1段だけ速くする(S46)。また、プロセッサA11がプログラムAの処理を完了するのと同じくらいに、プロセッサB12が処理を完了していた場合、すなわち情報I(n-1)が“0”でない場合には(S45, No)、クロック周波数を切替えずに、ステップS47へ進む。

【0066】

次に、プロセッサA11はプロセッサA11内に設けられたレジスタwait__countに“0”を代入し(S47)、プログラムAの実行を開始する(S48)。プロセッサA11は、プログラムAの実行を完了すると、上述した通信機能を用いてプロセッサB12から処理完了の通知があるか否かを判定する(S49)。プロセッサB12から処理完了の通知がなければ(S49, No)、wait__countの値をカウントアップし(S50)、ステップS49へ戻る。また、プロセッサB12から処理完了の通知があれば(S49, Yes)、情報I(n)にwait__countの値を代入して、情報I(n)を決定し(S51)、処理を終了する。なお、nはプロセッサA11がプログラムAの処理を行なった回数を示しており、この情報I(n)の値は次の処理においてステップS42で参照される。

【0067】

図11は、図10に示すプロセッサA11の処理手順を、C言語風の疑似プログラムによって表わしたものである。このプログラムにおいて、“B__is__done”はプロセッサB12によるプログラムBの実行状態を示しており、“0”のときはプログラムBの実行中であることを示し、“1”のときはプログラムBの実行が完了していることを示している。なお、各ステップの処理内容は、コメント文で詳細に説明されているので、ここでの説明は行なわない。

【0068】

以上説明したように、本実施の形態におけるマルチプロセッサシステムによれば、プロセッサA11とプロセッサB12とが並列に処理を実行する際、プロセッサA11の処理とプロセッサB12の処理とのいずれが先に完了するかによって、プロセッサA11のクロック周波数を切替えてクロック周波数が最適となるようにした。したがって、CMOS論理回路の消費電力がクロック周波数に比例することから、プロセッサA11の消費電力を削減することが可能となった。

【0069】

また、実施の形態1～3におけるマルチプロセッサシステムと同様に、プロセッサA11のクロック周波数を切替えながら、プロセッサA11の電源電圧を細かく制御するようにすれば、CMOS論理回路の消費電力が電源電圧の2乗に比

例することから、プロセッサ A 1 1 の消費電力をさらに削減することが可能となる。

【0070】

（実施の形態5）

実施の形態4のマルチプロセッサシステムにおいては、前回の処理においてプロセッサ A 1 1 がプロセッサ B 1 2 からのプログラム B の処理完了通知を待ったか否かによって、クロック周波数を変更するものであった。本発明の実施の形態5におけるマルチプロセッサシステムは、過去の複数回の処理においてプロセッサ A 1 1 がプロセッサ B 1 2 からのプログラム B の処理完了通知を待ったか否かによって、次の処理においてプロセッサ A 1 1 とプロセッサ B 1 2 とのいずれが早く処理を完了するかを予測して、クロックを制御するものである。

【0071】

本発明の実施の形態におけるマルチプロセッサシステムの概略構成は、図4に示す実施の形態4におけるマルチプロセッサシステムの概略構成と同じである。したがって、重複する構成および機能の詳細な説明は繰返さない。

【0072】

図12は、本実施の形態におけるプロセッサ A 1 1 の処理手順を説明するためのフローチャートである。まず、プロセッサ A 1 1 は、プロセッサ B 1 2 に対して起動要求を出力して、プロセッサ B 1 2 にプログラム B の実行を要求する（S 6 1）。そして、プロセッサ A 1 1 は、プロセッサ A 1 1 内に設けられた状態レジスタの値によって現在の状態を参照し（S 6 2）、プロセッサ A 1 1（プログラム A）の方がプロセッサ B 1 2（プログラム B）よりも早く処理を完了すると予測したか否かを判定する（S 6 3）。なお、状態レジスタは、後述する図14または図15に示す状態の番号が保持されるレジスタであり、この状態レジスタの値を更新することによって状態の遷移が行なわれる。

【0073】

プロセッサ A 1 1 が、プロセッサ A 1 1 の方がプロセッサ B 1 2 よりも早く処理を完了すると予測した場合には（S 6 3, Y e s）、クロック周波数制御部 1 5 を制御して、クロック周波数を1段だけ遅くする（S 6 4）。また、プロセッ

サ A 1 1 が、プロセッサ A 1 1 の方がプロセッサ B 1 2 よりも早く処理を完了すると予測しなかった場合には (S 6 3, N o)、プロセッサ B 1 2 の方がプロセッサ A 1 1 よりも早く処理を完了すると予測したか否かを判定する (S 6 5)。

【0 0 7 4】

プロセッサ B 1 2 の方がプロセッサ A 1 1 よりも早く処理を完了すると予測した場合には (S 6 5, Y e s)、クロック制御部 1 5 を制御して、クロック周波数を 1 段だけ速くする (S 6 6)。また、プロセッサ B 1 2 の方がプロセッサ A 1 1 よりも早く処理を完了すると予測しなかった場合には (S 6 5, N o)、クロック周波数を切替えずに、ステップ S 6 7 へ進む。

【0 0 7 5】

次に、プロセッサ A 1 1 はプロセッサ A 1 1 内に設けられたレジスタ `wait__count` に “0” を代入し (S 6 7)、プログラム A の実行を開始する (S 6 8)。プロセッサ A 1 1 は、プログラム A の実行を完了すると、上述した通信機能を用いてプロセッサ B 1 2 から処理完了の通知があるか否かを判定する (S 6 9)。プロセッサ B 1 2 から処理完了の通知がなければ (S 6 9, N o)、`wait__count` の値をカウントアップし (S 7 0)、ステップ S 6 9 へ戻る。また、プロセッサ B 1 2 から処理完了の通知があれば (S 6 9, Y e s)、`wait__count` の値により状態を更新し (S 7 1)、処理を終了する。なお、この状態の更新については後述する。

【0 0 7 6】

図 1 3 は、図 1 2 に示すプロセッサ A 1 1 の処理手順を、C 言語風の疑似プログラムによって表わしたものである。このプログラムにおいて、“`B__is__done`” はプロセッサ B 1 2 によるプログラム B の実行状態を示しており、“0” のときはプログラム B の実行中であることを示し、“1” のときはプログラム B の実行が完了していることを示している。また、“`wait__count`” は、プロセッサ A 1 1 がどれだけプロセッサ B 1 2 の処理完了を待ったかを示している。この“`wait__count`” の値によって、後述する状態の遷移が決定される。なお、各ステップの処理内容は、コメント文で詳細に説明されているので、ここでの説明は行なわない。

【 0 0 7 7 】

図 1 4 は、次の処理においてプロセッサ A 1 1 とプロセッサ B 1 2 とのいずれが早く処理を完了するかを予測するための状態遷移図である。“現在の状態から予測値を得る”および“状態を更新する”の 2 つの手続は、発見的手法により様々な方法が考えられる。図 1 4 は、その一例を示しており、楕円は状態を表わし、矢印は状態の更新を表わしている。この状態遷移図は、プロセッサ A 1 1（プログラム A）とプロセッサ B 1 2（プログラム B）とのいずれが早く処理を完了したかによって、状態をどのように遷移させるかを規定しており、それぞれの状態には予測値が割当てられている。

【 0 0 7 8 】

たとえば、現在の状態が“状態 1”にあるとき、“現在の状態から予測値を取出す”という操作によって、“状態 1”に割当てられている“プログラム A が早く処理を完了すると予測”という予測値が得られる。また、現在の状態が“状態 1”にあるとき、プログラム B が早く完了した場合には“状態を更新する”という操作によって、“状態 1”から“状態 2”へ状態が更新される。

【 0 0 7 9 】

図 1 4 に示す状態遷移図に準拠して図 1 2 のステップ S 7 1 の処理が行なわれる場合、あるしきい値 `WAIT_LIMIT` (≥ 0) が予め定められており、`wait_count` の値が `WAIT_LIMIT` を超えたときに“プログラム A が早く完了した”と判断し、そうでないならば“プログラム B が早く完了した”と判断して、現在の状態レジスタの値を参照して図 1 4 に示す状態遷移が生じるように状態を更新する。

【 0 0 8 0 】

図 1 4 に示す状態遷移図においては、“プログラム A が早く処理を完了したら、クロック周波数を低くしても、やはりプログラム A が早く処理を完了する傾向にある”という戦略を表現している。これに対して、“プログラム A が早く処理を完了したら、クロック周波数を低くすることになるので、次はプログラム B が早く処理を完了する傾向がある”という戦略が考えられる。図 1 5 は、この戦略に基づく状態遷移図を示しており、プログラム A の処理時間とプログラム B の処

理時間とがちょうどつりあう状態も考慮している。

【0081】

図15に示す状態遷移図に準拠して図12のステップS71の処理が行なわれる場合、ある2つのしきい値WAIT_LIMIT1、WAIT_LIMIT2（ただし、 $0 \leq \text{WAIT_LIMIT1} < \text{WAIT_LIMIT2}$ ）が予め定められており、wait_countの値がWAIT_LIMIT2を超える場合には“プログラムAが早く完了した”と判断し、wait_countの値がWAIT_LIMIT1を超えかつWAIT_LIMIT2以下である場合には“プログラムA、Bはほぼ同時に処理を完了した”と判断し、さらにwait_countの値がWAIT_LIMIT1以下である場合には“プログラムBが早く完了した”と判断し、現在の状態レジスタの値を参照して図15に示す状態遷移が生じるように状態を更新する。

【0082】

以上説明したように、本実施の形態におけるマルチプロセッサシステムによれば、プロセッサA11とプロセッサB12とが並列に処理を実行する際、プロセッサA11の処理とプロセッサB12の処理とのいずれが先に完了するかを予測して、プロセッサA11のクロック周波数を切替えてクロック周波数が最適となるようにした。したがって、CMOS論理回路の消費電力がクロック周波数に比例することから、プロセッサA11の消費電力を削減することが可能となった。

【0083】

また、実施の形態1～3におけるマルチプロセッサシステムと同様に、プロセッサA11のクロック周波数を切替えながら、プロセッサA11の電源電圧を細かく制御するようにすれば、CMOS論理回路の消費電力が電源電圧の2乗に比例することから、プロセッサA11の消費電力をさらに削減することが可能となる。

【0084】

（実施の形態6）

図16は、本発明の実施の形態6におけるマルチプロセッサシステムの概略構成を示すブロック図である。本実施の形態におけるマルチプロセッサシステムは

、図 1 に示す実施の形態 1 におけるマルチプロセッサシステムと比較して、プロセッサ A 1 1 によるプログラム A の処理時間を計測するカウンタ 2 1 と、プロセッサ B 1 2 によるプログラム B の処理時間を計測するカウンタ 2 2 とが追加された点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。

【 0 0 8 5 】

カウンタ 2 1 および 2 2 は、共通のクロック（カウンタ用クロック）をカウントすることによってカウントアップする。このカウント用クロックには、プロセッサ A 1 1 に供給されるクロックおよびプロセッサ B 1 2 に供給されるクロックよりも周波数が小さいものが使用されるのが好ましい。なぜなら、低消費電力化のためである。

【 0 0 8 6 】

カウンタ 2 1 は、プロセッサ A 1 1 がプログラム A の実行を開始するときにリセットされ、プログラム A の実行中にカウントアップする。そして、プロセッサ A 1 1 がプログラム A の実行を終了するときに、カウンタ 2 1 のカウント値が処理時間見積値 T a としてメモリ 1 3 に転送される。また、カウンタ 2 2 は、プロセッサ B 1 2 がプログラム B の実行を開始するときにリセットされ、プログラム B の実行中にカウントアップする。そして、プロセッサ B 1 2 がプログラム B の実行を終了するときに、カウンタ 2 2 のカウント値が処理時間見積値 T b としてメモリ 1 4 に転送される。

【 0 0 8 7 】

プロセッサ A 1 1 は、次回のプロセッサ A 1 1 によるプログラム A の実行時において、メモリ 1 3 に格納されたプログラム A の処理時間見積値 T a およびメモリ 1 4 に格納されたプログラム B の処理時間見積値 T b を参照して、プロセッサ A 1 1 に入力されるクロック周波数を決定し、クロック周波数制御部 1 5 にクロック制御指令を発行する。

【 0 0 8 8 】

本実施の形態におけるマルチプロセッサシステムの処理手順は、図 3 に示す実施の形態 1 におけるマルチプロセッサシステムの処理手順と同じであるので、詳

細な説明は繰返さない。

【 0 0 8 9 】

以上説明したように、本実施の形態におけるマルチプロセッサシステムによれば、プロセッサ A 1 1 とプロセッサ B 1 2 とが並列に処理を実行する際、前回のプログラム A の処理時間の計測値とプログラム B の処理時間の計測値とを参照し、プロセッサ A 1 1 がプロセッサ B 1 2 よりも先に処理が完了する場合には、プロセッサ A 1 1 を低速動作させるようにした。したがって、CMOS 論理回路の消費電力がクロック周波数に比例することから、プロセッサ A 1 1 の消費電力を削減することが可能となった。

【 0 0 9 0 】

また、プロセッサ A 1 1 を低速動作させることにより、プロセッサ A 1 1 の電源電圧を下げるようにした。したがって、CMOS 論理回路の消費電力が電源電圧の 2 乗に比例することから、プロセッサ A 1 1 の消費電力をさらに削減することが可能となった。

【 0 0 9 1 】

今回開示された実施の形態は、すべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【 0 0 9 2 】

【発明の効果】

請求項 1 に記載のマルチプロセッサシステムによれば、クロック周波数制御手段が、第 1 のプロセッサの処理時間と第 2 のプロセッサの処理時間との比率に応じて、第 1 のプロセッサに入力されるクロックの周波数を制御するので、第 1 のプロセッサに入力されるクロックの周波数を最適にすることができ、第 1 のプロセッサの消費電力を削減することが可能となった。

【 0 0 9 3 】

請求項 2 に記載のマルチプロセッサシステムによれば、第 1 のプロセッサに入力されるクロックの周波数の低減に応じて、電源電圧を下げるができるため

、さらに第1のプロセッサの消費電力を削減することが可能となった。

【0094】

請求項3に記載のマルチプロセッサシステムによれば、周波数決定手段が、第1のプロセッサの処理時間見積値と第2のプロセッサの処理時間見積値との比率に応じて、第1のプロセッサに入力されるクロックの周波数を変更するので、クロック周波数を変更するための処理を簡略化することが可能となった。

【0095】

請求項4に記載のマルチプロセッサシステムによれば、周波数決定手段が、格納手段に格納された比率に応じて、第1のプロセッサに入力されるクロックの周波数を変更するので、クロック周波数を変更するための処理を簡略化することが可能となった。

【0096】

請求項5に記載のマルチプロセッサシステムによれば、周波数決定手段が、前回の第1のプロセッサおよび第2のプロセッサの処理状況に応じて、第1のプロセッサに入力されるクロックの周波数を変更するので、プロセッサの処理時間が実行毎に異なるような場合であっても、最適なクロック周波数を決定することが可能となった。

【0097】

請求項6に記載のマルチプロセッサシステムによれば、周波数決定手段が、予測手段による予測結果に応じて、第1のプロセッサに入力されるクロックの周波数を変更するので、最適なクロック周波数を決定することが可能となった。

【0098】

請求項7に記載のマルチプロセッサシステムによれば、第1のプロセッサが早く処理を完了したら、クロック周波数を低くしても、やはり第1のプロセッサが早く処理を完了する傾向にあるという戦略を表わすことができ、最適なクロック周波数を決定することが可能となった。

【0099】

請求項8に記載のマルチプロセッサシステムによれば、第1のプロセッサが早く処理を完了したら、クロック周波数を低くすることになるので、次は第2のプ

ロセッサが早く処理を完了する傾向にあるという戦略を表わすことができ、最適なクロック周波数を決定することが可能となった。

【0 1 0 0】

請求項 9 に記載のマルチプロセッサシステムによれば、周波数決定手段が、第 1 の計測手段による計測結果および第 2 の計測手段による計測結果に応じてクロック周波数制御手段を制御するので、最適なクロック周波数を決定することが可能となった。

【0 1 0 1】

請求項 1 0 に記載のマルチプロセッサシステムの制御方法によれば、第 1 のプロセッサの処理時間と第 2 のプロセッサの処理時間との比率に応じて、第 1 のプロセッサに入力されるクロックの周波数を制御するので、第 1 のプロセッサに入力されるクロックの周波数を最適にすることができ、第 1 のプロセッサの消費電力を削減することが可能となった。

【0 1 0 2】

請求項 1 1 に記載のマルチプロセッサシステムの制御方法によれば、第 1 のプロセッサに入力されるクロックの周波数の低減に応じて、電源電圧を下げることで、さらに第 1 のプロセッサの消費電力を削減することが可能となった。

【図面の簡単な説明】

【図 1】 本発明の実施の形態 1 におけるマルチプロセッサシステムの概略構成を示すブロック図である。

【図 2】 クロック周波数制御部 1 5 の詳細を説明するための図である。

【図 3】 本発明の実施の形態 1 におけるプロセッサ A 1 1 の処理手順を説明するためのフローチャートである。

【図 4】 図 3 に示すプロセッサ A 1 1 の処理手順を C 言語風の疑似プログラムによって表わした図である。

【図 5】 本発明の実施の形態 2 におけるマルチプロセッサシステムの概略構成を示すブロック図である。

【図 6】 本発明の実施の形態 2 におけるプロセッサ A 1 1 の処理手順を説

明するためのフローチャートである。

【図 7】 本発明の実施の形態 3 におけるマルチプロセッサシステムの概略構成を示すブロック図である。

【図 8】 本発明の実施の形態 3 におけるプロセッサ A 1 1 の処理手順を説明するためのフローチャートである。

【図 9】 本発明の実施の形態 4 におけるマルチプロセッサシステムの概略構成を示すブロック図である。

【図 1 0】 本発明の実施の形態 4 におけるプロセッサ A 1 1 の処理手順を説明するためのフローチャートである。

【図 1 1】 図 1 0 に示すプロセッサ A 1 1 の処理手順を C 言語風の疑似プログラムによって表わした図である。

【図 1 2】 本発明の実施の形態 5 におけるプロセッサ A 1 1 の処理手順を説明するためのフローチャートである。

【図 1 3】 図 1 2 に示すプロセッサ A 1 1 の処理手順を C 言語風の疑似プログラムによって表わした図である。

【図 1 4】 プロセッサ A 1 1 とプロセッサ B 1 2 とのいずれが早く処理を完了するかを予測するための状態遷移の一例を示す図である。

【図 1 5】 プロセッサ A 1 1 とプロセッサ B 1 2 とのいずれが早く処理を完了するかを予測するための状態遷移の他の一例を示す図である。

【図 1 6】 本発明の実施の形態 6 におけるマルチプロセッサシステムの概略構成を示すブロック図である。

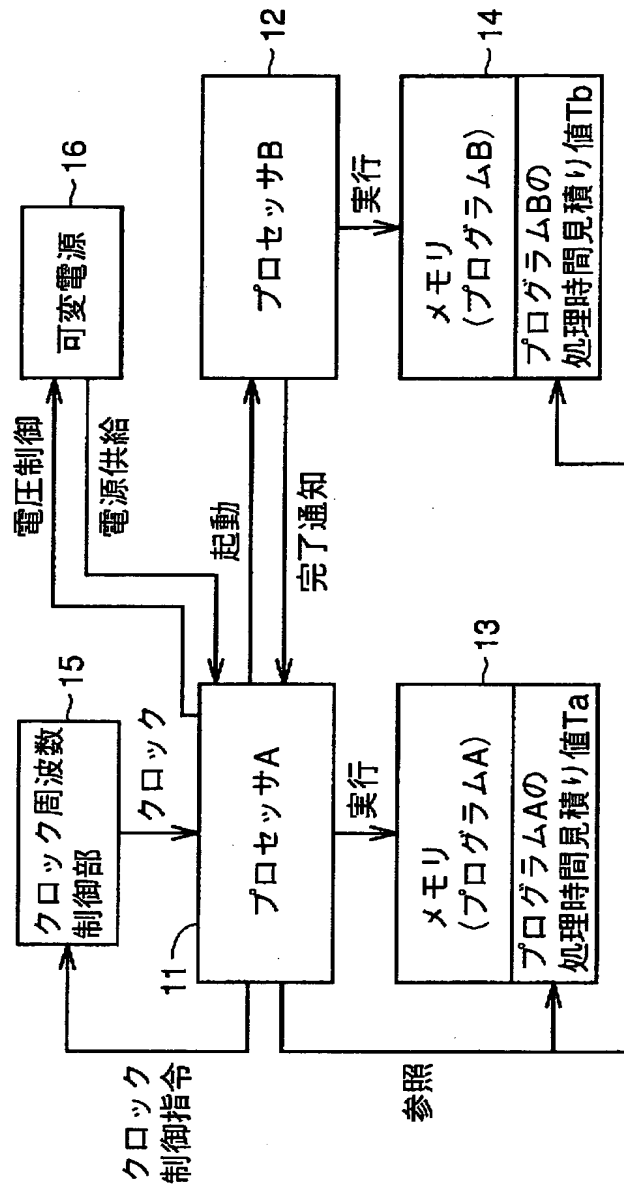
【図 1 7】 従来のマルチプロセッサシステムの概略構成を示すブロック図である。

【符号の説明】

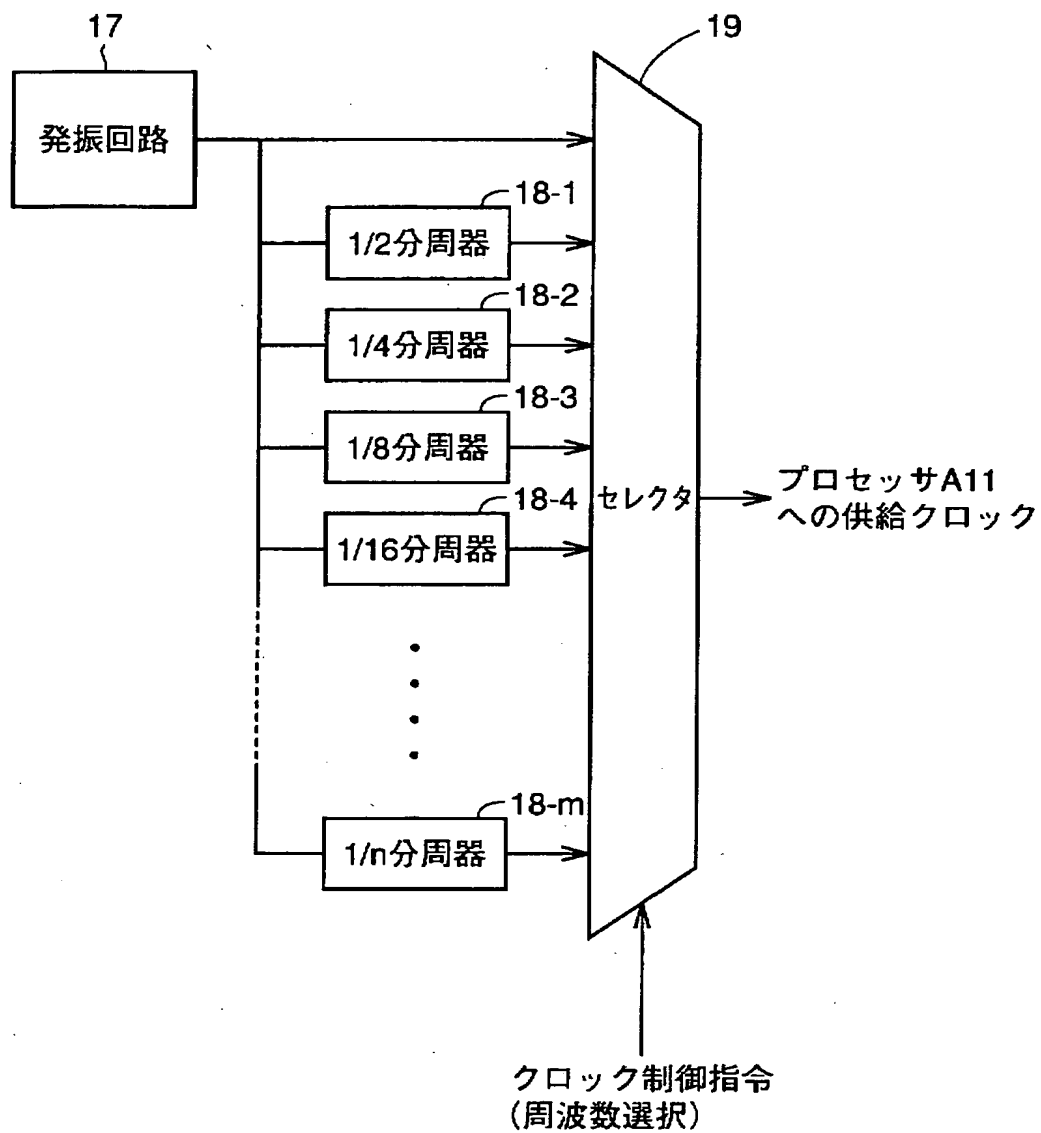
1 1 プロセッサ A、1 2 プロセッサ B、1 3、1 4 メモリ、1 5 クロック周波数制御部、1 6 可変電源、1 7 発振回路、1 8 - 1 ~ 1 8 - m 分周器、1 9 セレクタ、2 1 プログラム A の処理時間計測カウンタ、2 2 プログラム B の処理時間計測カウンタ。

【書類名】 図面

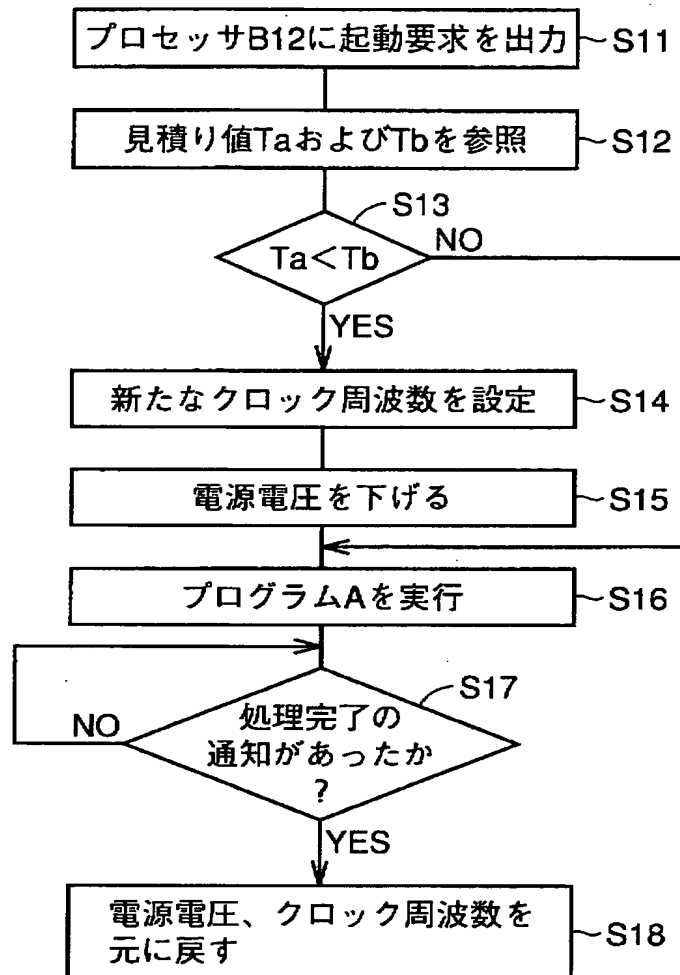
【図 1】



【図 2】



【図 3】



【図 4】

```

main_procedure()
{
    B_is_done=0;
    invokeB();

    set_clock_frequency(Ta/Tb);
    set_power(table);
    do_something();

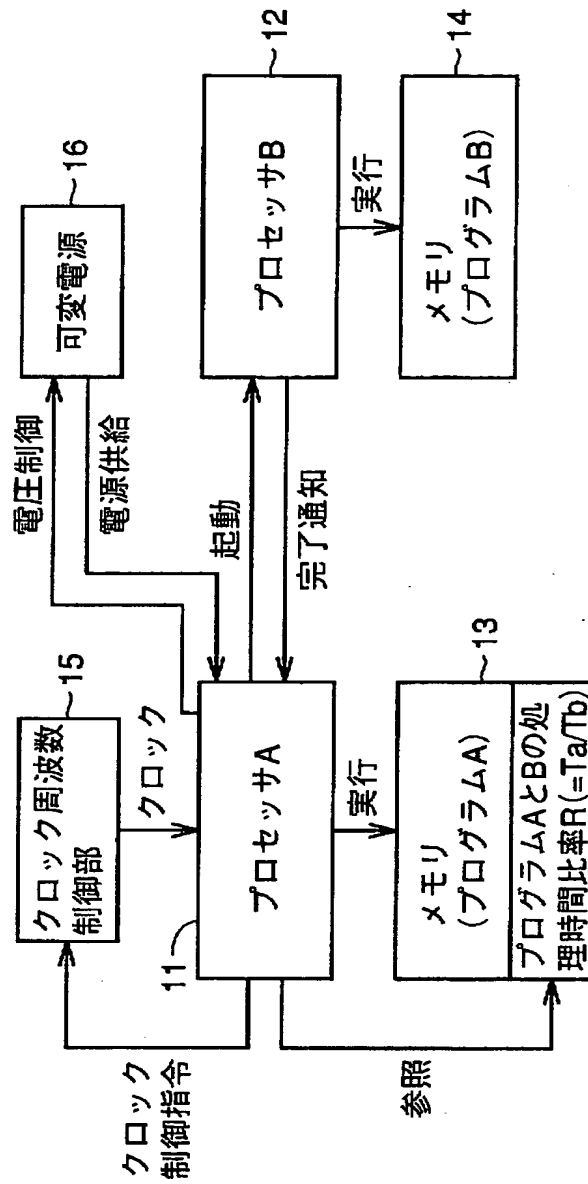
    while (B_is_done)
        /* do nothing */;
}

interrupt_from_B()
{
    restore_power();
    restore_clock_frequency();
    B_is_done=1;
}

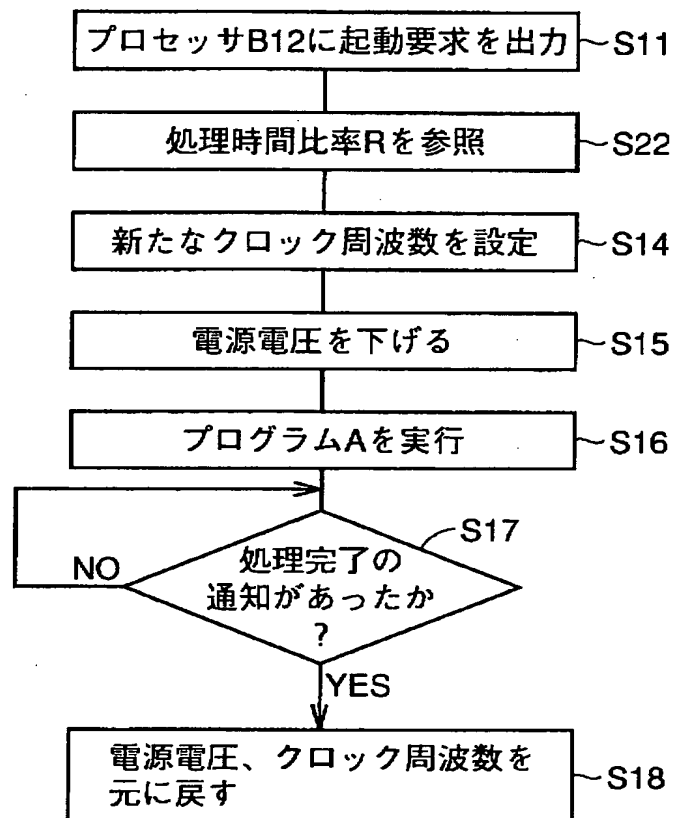
```

/* プログラムAのメイン処理部 */
 /* プログラムBを起動する。処理時間見積りはTb */
 /* クロック周波数をTa/Tbを限度として遅くする。 */
 /* テーブルを参照して電圧を下げる */
 /* プログラムAで行うべき処理。処理時間見積りはTa */
 /* プログラムBの完了を待つ */
 /* プログラムAの割り込み処理ルーチン */
 /* 電圧を元に戻す */
 /* クロック周波数を元に戻す */

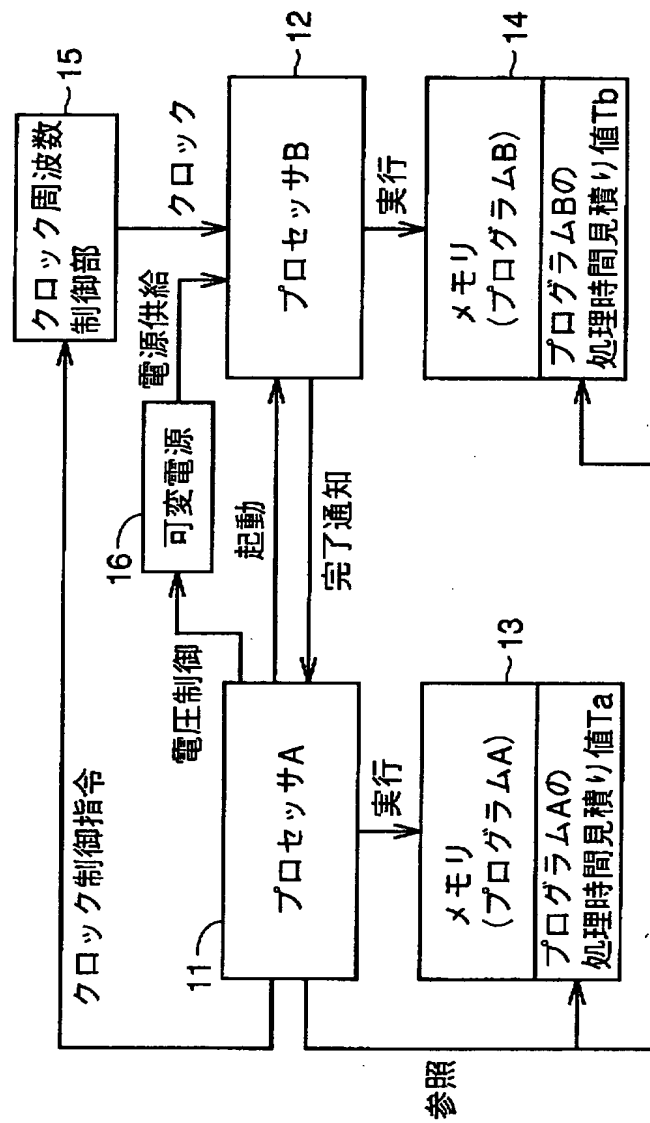
【図5】



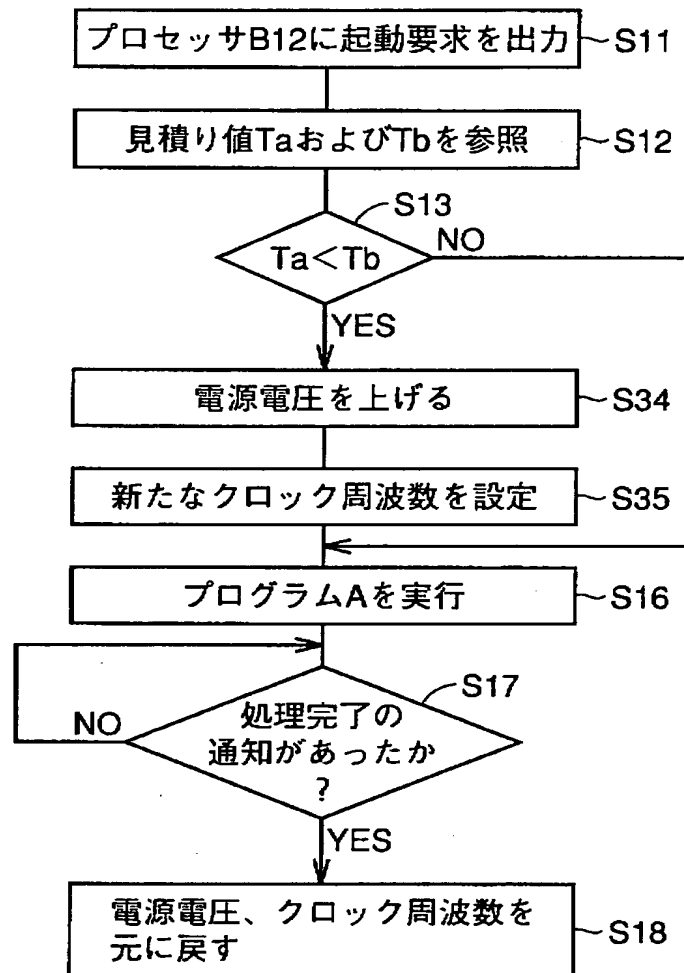
【図 6】



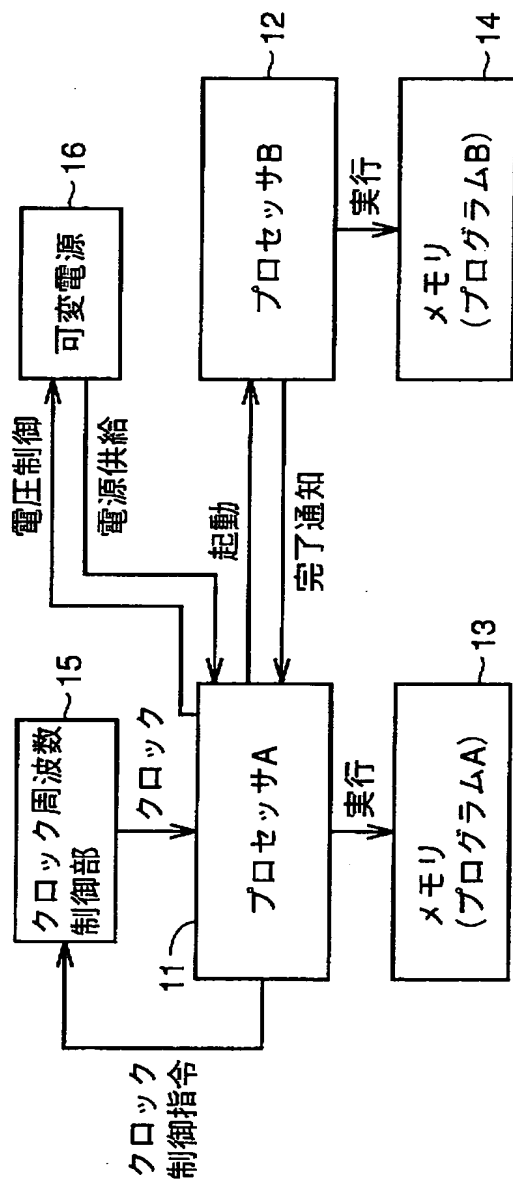
【図 7】



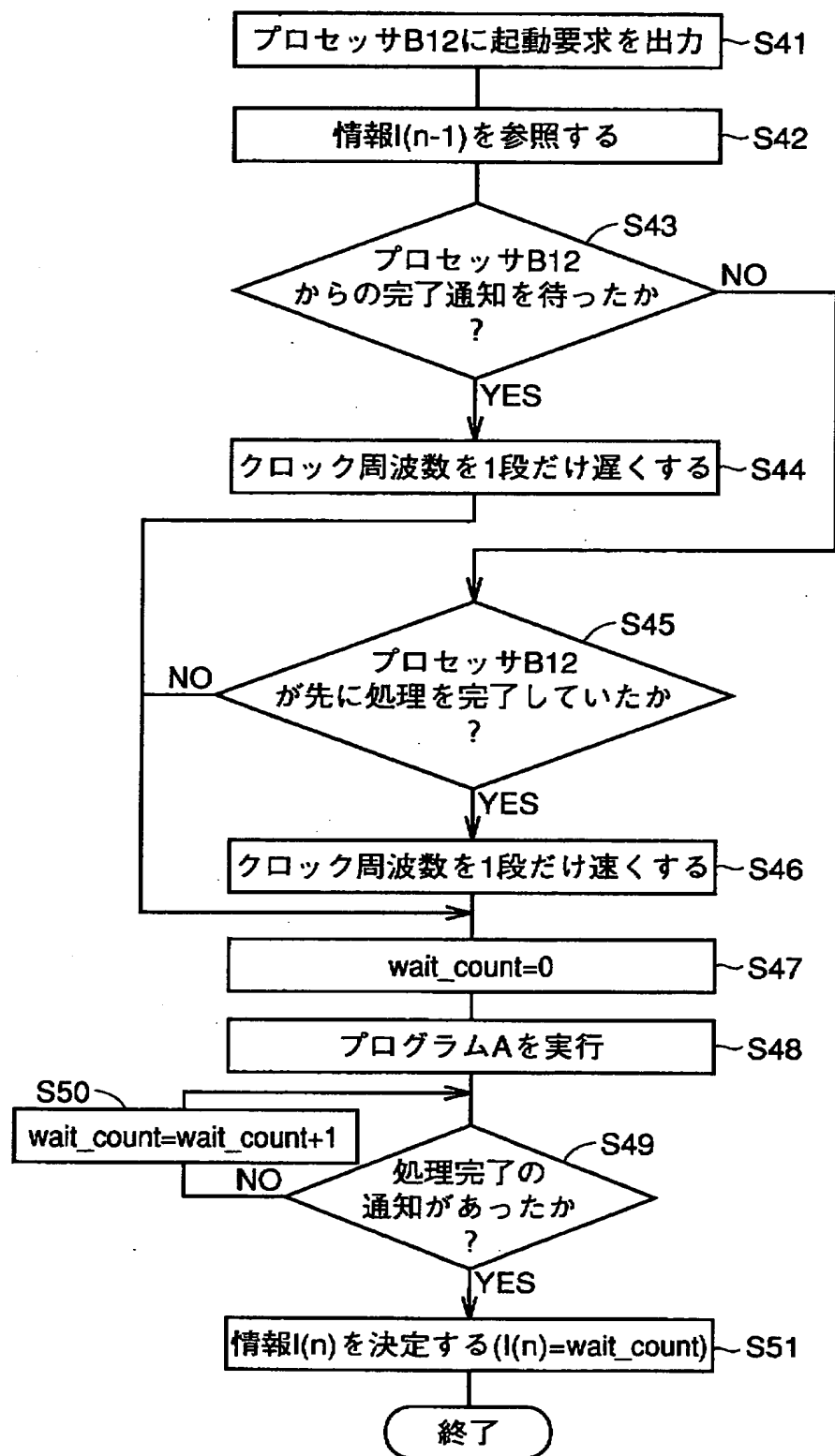
【図 8】



【図9】



【図 1 0】



【図 1 1】

```

main_procedure()
{
    B_is_done=0;
    invokeB();

    /* プログラムAのメイン処理部 */

    /* プログラムBを起動する。 */

    /* 前回の処理にてプログラムBの完了を待った場合には */
    /* クロック周波数を1段だけ遅くする。 */
    /* プログラムBの方が早く完了した場合には */
    /* クロック周波数を1段だけ速くする。 */
    /* プログラムAとプログラムBが同じくいらに完了した場合には */
    /* 何もせずに現在のクロック周波数を維持する */

    if(wait_count>WAIT_LIMIT)
        set_clock_faster();
    else if(wait_count==0)
        set_clock_slower();
    else
        /* do nothing */;
    wait_count=0;

    do_something();

    /* プログラムAで行うべき処理。 */

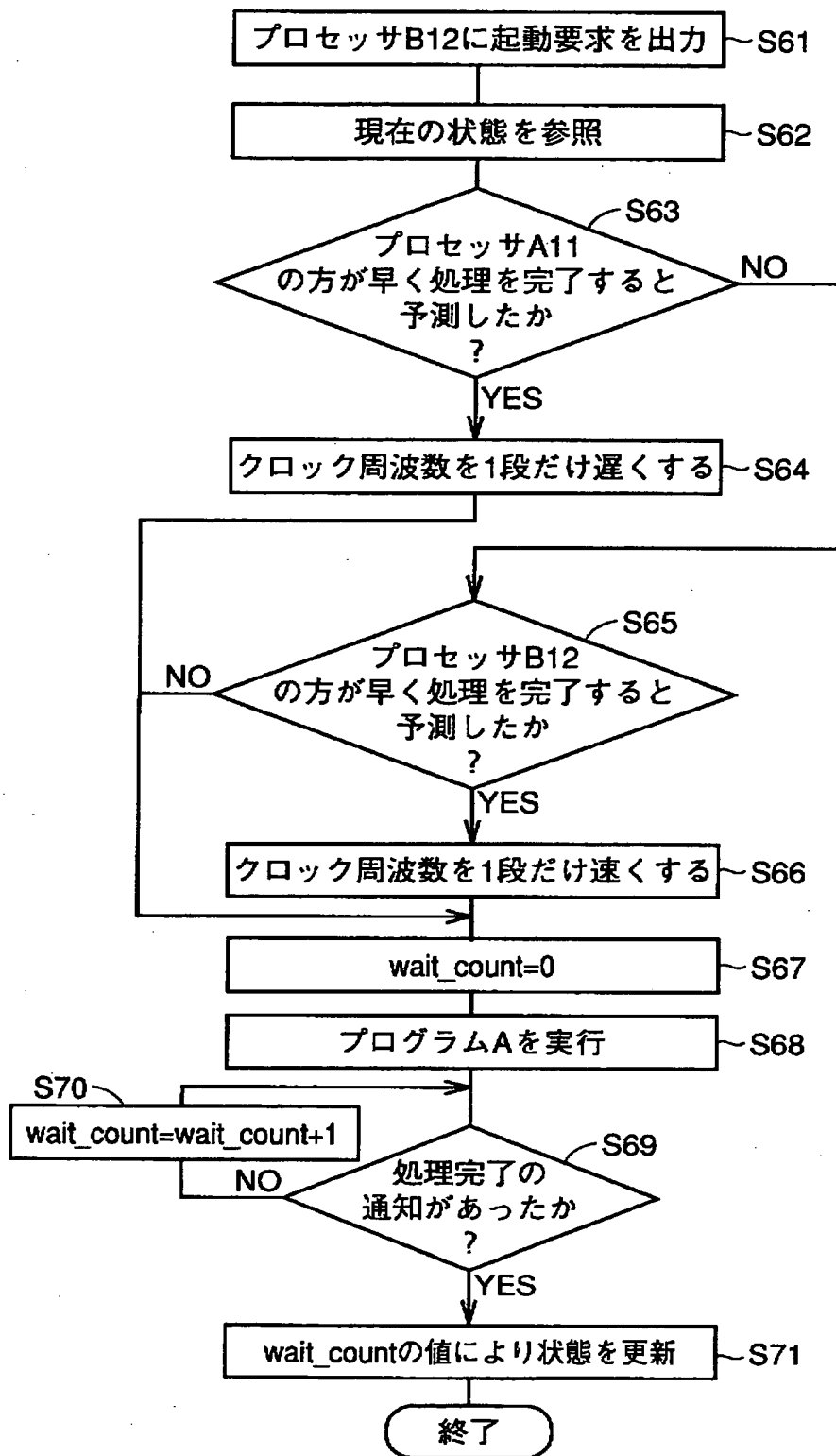
    while (B_is_done==0)
        wait_count++;
    }

    interrupt_from_B()
    {
        B_is_done=1;
    }

    /* プログラムAの割り込み処理ルーチン */

```

【図 1 2】



【図 1 3】

```

main_procedure()      /* プログラムAのメイン処理部 */
{
    B_is_done=0;
    invokeB();          /* プログラムBを起動する。 */

    予測値=現在の状態から予測値を得る;
    if(予測値==プログラムAが早く処理を完了すると予測)
        set_clock_faster(); /* クロック周波数を1段だけ遅くする。 */
    else(予測値==プログラムBが早く処理を完了すると予測)
        set_clock_slower(); /* クロック周波数を1段だけ速くする。 */
    else /* プログラムAとプログラムBが同じくらいに完了すると予測 */
        /* do nothing */;    /* 何もせず現在のクロック周波数を維持する */
    wait_count=0;

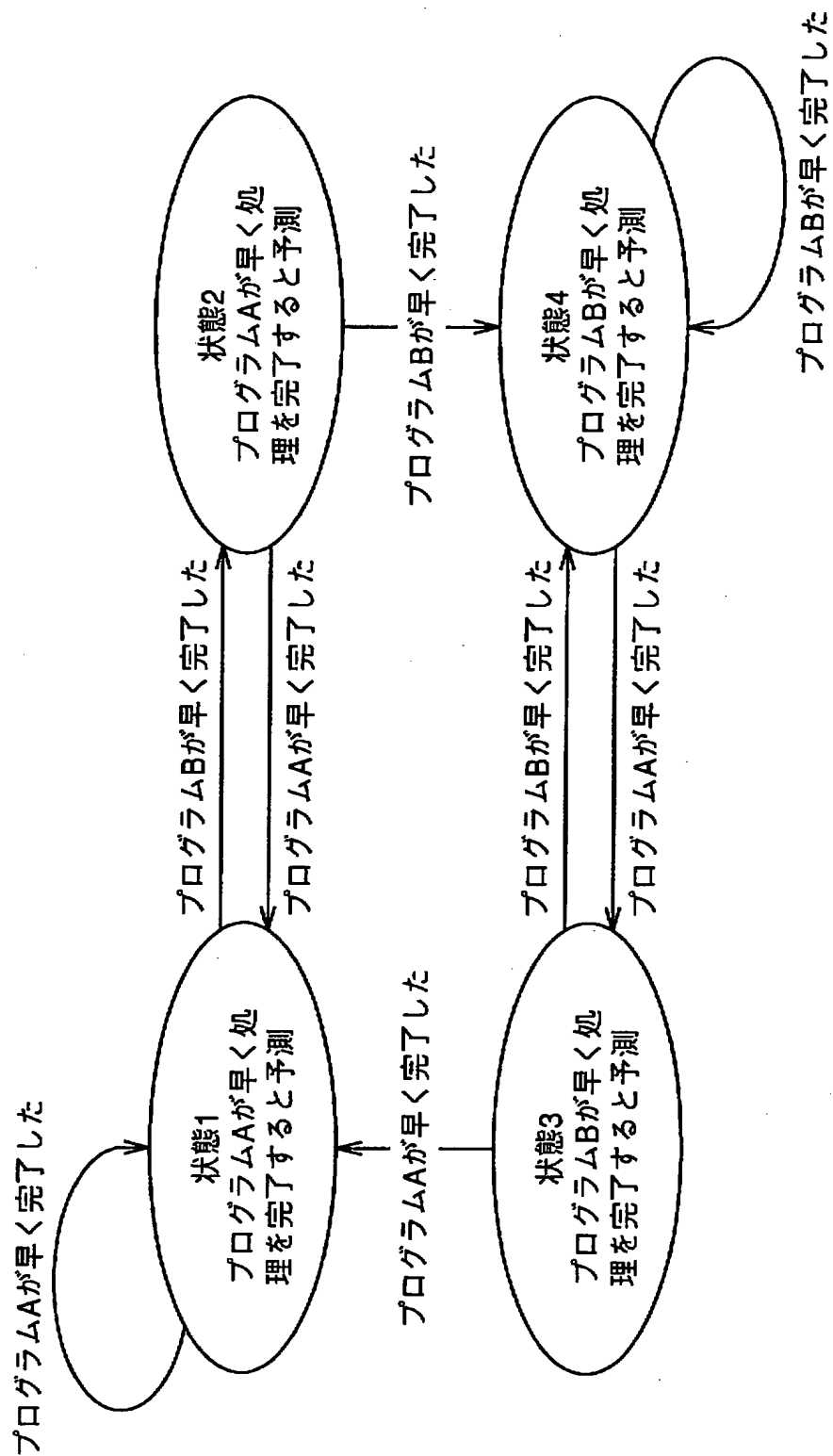
    do_something();      /* プログラムAで行うべき処理。 */

    while (B_is_done==0) /* プログラムBの完了を待つ */
        wait_count++;
    wait_countの値より、状態を更新する;
}

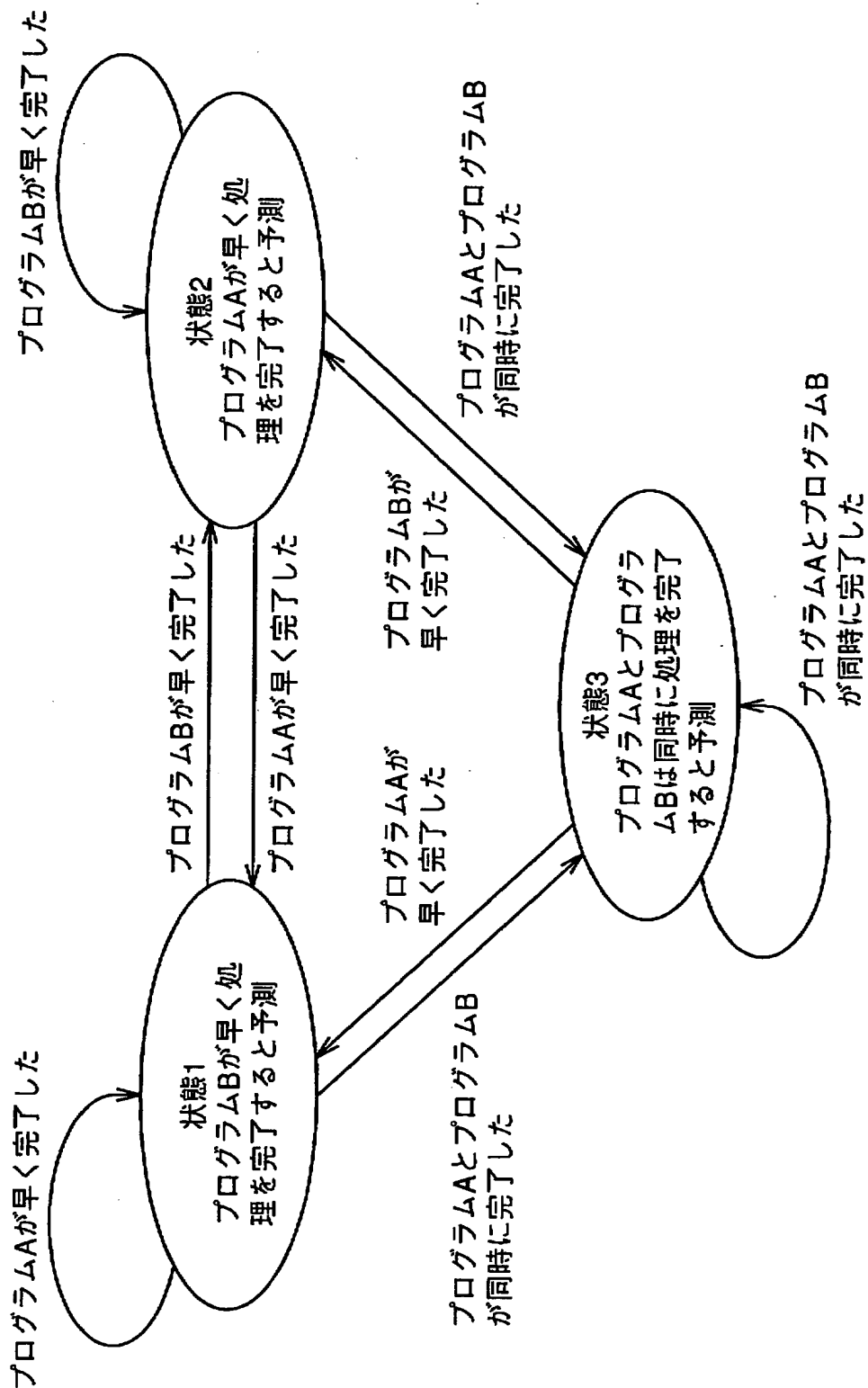
interrupt_from_B()
{
    B_is_done=1;
}

```

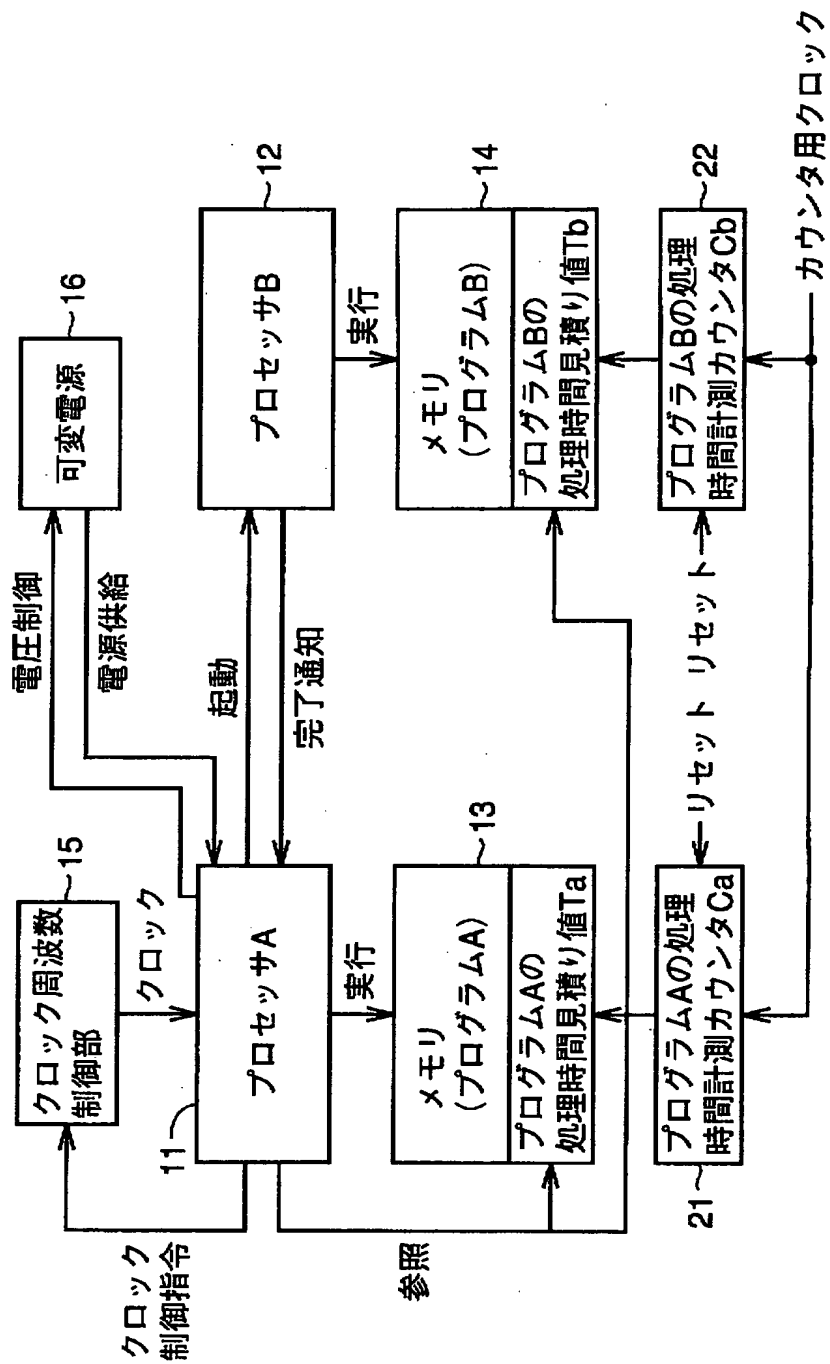
【図14】



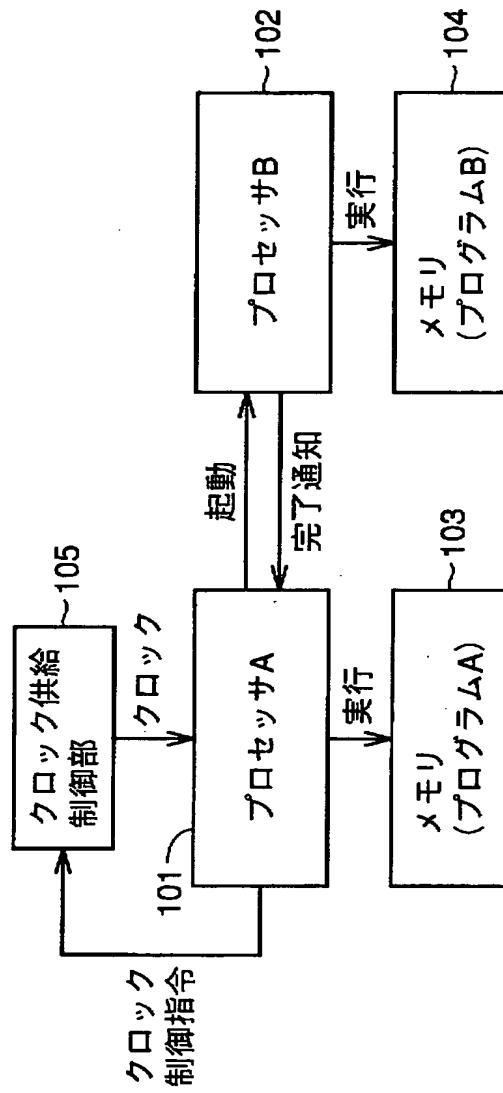
【図 15】



【図16】



【図 1 7】



【書類名】 要約書

【要約】

【課題】 消費電力を大きく削減することが可能なマルチプロセッサシステムを提供すること。

【解決手段】 プロセッサ A 1 1 は、プログラム A の処理時間見積値 T_a とプログラム B の処理時間見積値 T_b との比率に応じて、クロック周波数制御部 1 5 を制御してプロセッサ A 1 1 に入力されるクロックの周波数を変更する。また、プロセッサ A 1 1 は、プロセッサ A 1 1 に入力されるクロックの周波数に応じて可変電源 1 6 を制御して、プロセッサ A 1 1 に供給される電源電圧を変更する。したがって、プロセッサ A 1 1 の消費電力を大きく削減することが可能となる。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000006013]

1. 変更年月日 1990年 8月24日
[変更理由] 新規登録
住 所 東京都千代田区丸の内2丁目2番3号
氏 名 三菱電機株式会社